# Learning to Model and Plan for Wheeled Mobility on Vertically Challenging Terrain

**Anonymous Author(s)**
Affiliation
Address
email

**Abstract:** Most autonomous navigation systems assume wheeled robots are rigid bodies and their 2D planar workspaces can be divided into free spaces and obstacles. However, recent wheeled mobility research, showing that wheeled platforms have the potential of moving over vertically challenging terrain (e.g., rocky outcroppings, rugged boulders, and fallen tree trunks), invalidate both assumptions. Navigating off-road vehicle chassis with long suspension travel and low tire pressure in places where the boundary between obstacles and free spaces is blurry requires precise 3D modeling of the interaction between the chassis and the terrain, which is complicated by suspension and tire deformation, varying tire-terrain friction, vehicle weight distribution and momentum, etc. In this paper, we present a learning approach to model wheeled mobility, i.e., in terms of vehicle-terrain forward dynamics, and plan feasible, stable, and efficient motion to drive over vertically challenging terrain without rolling over or getting stuck. We present physical experiments on two wheeled robots and show that planning using our learned model can achieve up to 60% improvement in navigation success rate and 46% reduction in unstable chassis roll and pitch angles.

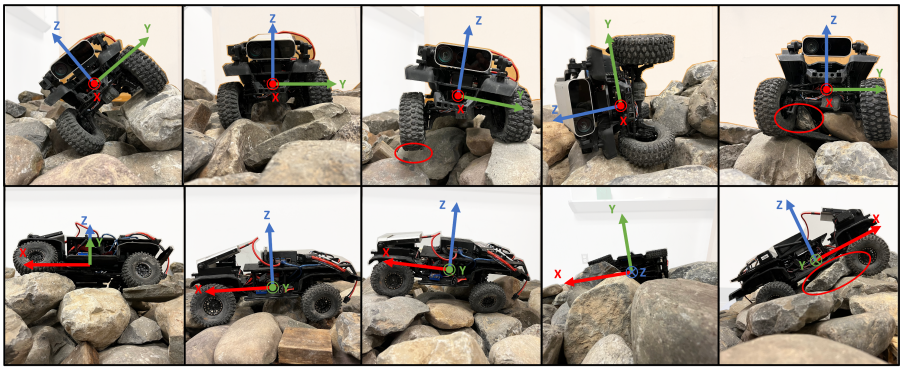

Figure 1: Front and side view (1st and 2nd row) of a wheeled robot navigating vertically challenging terrain: (from left to right) large roll angle, stable chassis, suspended wheel, roll-over, and get-stuck.

## 1 Introduction

Wheeled robots, arguably the most commonly used mobile robot type, have autonomously moved from one point to another in a collision-free and efficient manner in the real world, e.g., transporting materials in factories or warehouses [1], vacuuming our homes or offices [2], and delivering food or packages on sidewalks [3]. Thanks to their simple motion mechanism, most wheeled robots are treated as rigid bodies moving through planar workspaces. After tessellating their 2D workspaces into obstacles and free spaces, classical planning algorithms plan feasible paths in the free spaces that are free of collisions with the obstacles [4, 5, 6].

Submitted to the 7th Conference on Robot Learning (CoRL 2023). Do not distribute.

However, recent advances in wheeled mobility have shown that even conventional wheeled robots (i.e., without extensive hardware modification such as active suspensions [7, 8, 9] or adhesive materials [10]) have previously unrealized potential to move over vertically challenging terrain (e.g., in mountain passes with large boulders or dense forests with fallen trees) [11, 12, 13], where vehicle motion is no longer constrained to a 2D plane [14] (Figure 1). In those environments, neither assumptions of rigid vehicle chassis and clear delineation between obstacles and free spaces in a simple 2D plane are valid [15, 16, 17, 18]. Thanks to the long suspension travel and reduced tire pressure, off-road vehicle chassis are able to drive *over* obstacles (rather than to *avoid* them) and experience significant deformation to conform with the irregular terrain underneath the robot, which will be otherwise deemed as non-traversable according to conventional navigation systems. Therefore, autonomously navigating wheeled robots in vertically challenging terrain without rolling over or getting stuck requires a precise understanding of the 3D vehicle-terrain interaction.

In this paper, we investigate learning approaches to model vehicle-terrain interactions and plan vehicle trajectories to drive wheeled robots on vertically challenging terrain. Considering the difficulty in analytically modeling and computing vehicle poses using complex vehicle dynamics [19, 20, 21] in real time, we adopt a data-driven approach to model the forward vehicle-terrain dynamics based on terrain elevation maps along potential future trajectories. We develop a Wheeled Mobility on Vertically Challenging Terrain (WM-VCT) planner, which uses our learned model's output in a novel cost function and produces feasible, stable, and efficient motion plans to autonomously navigate wheeled robots on vertically challenging terrain. We present extensive physical experiment results on two wheeled robot platforms and compare our learning approach against four existing baselines and show that our learned model can achieve up to 60% improvement in navigation success rate and 46% reduction in unstable chassis roll and pitch angles.

## 2 Approach

The difficulties in navigating a wheeled mobile robot on vertically challenging terrain are two fold: (1) the high variability of vehicle poses due to the irregular terrain underneath the robot may overturn the vehicle (rolling-over, 4th column in Figure 1); (2) not being able to identify that a certain terrain patch is beyond the robot's mechanical limit and therefore needs to be circumvented may get the robot stuck (immobilization, 5th column in Figure 1). Therefore, this work takes a structured learning approach to address both challenges by learning a vehicle-terrain forward dynamics model based on the vertically challenging terrain underneath the vehicle, using it to rollout sampled receding-horizon trajectories, and minimizing a cost function to reduce the chance of rolling-over and immobilization and to move the vehicle toward the goal.

### 2.1 Motion Planning Problem Formulation

Consider a discrete vehicle dynamics model of the form $\mathbf{x}_{t+1} = f(\mathbf{x}_t, \mathbf{u}_t)$, where $\mathbf{x}_t \in X$ and $\mathbf{u}_t \in U$ denote the state and input space respectively. In the normal case of 2D navigation planning (Figure 2 left), $X \subset \mathbb{SE}(2)$ and $X = X_{\text{free}} \cup X_{\text{obs}}$, where $X_{\text{free}}$ and $X_{\text{obs}}$ denote free spaces and obstacle regions. $\mathbf{x}_t$ includes translation along the $\mathbf{x}$ and $\mathbf{y}$ axis ($x$ and $y$) and the rotation along the $\mathbf{z} = \mathbf{x} \times \mathbf{y}$ axis (yaw) of a fixed global coordinate system. For input, $U \subset \mathbb{R}^2$ and $\mathbf{u}_t = (v_t, \omega_t)$, where $v_t$ and $\omega_t$ are the linear and angular velocity. Finally, let $X_{\text{goal}} \subset X$ denote the goal region. The motion planning problem for the conventional 2D navigation case is to find a control function $u : \{t\}_{t=0}^{T-1} \rightarrow U$ that produces an optimal path $\mathbf{x}_t \in X_{\text{free}}, \forall t \in \{t\}_{t=0}^{T}$ from an initial state $\mathbf{x}_0 = \mathbf{x}_{\text{init}}$ to the goal region $\mathbf{x}_T \in X_{\text{goal}}$ that follows the system dynamics $f(\cdot, \cdot)$ and minimizes a given cost function $c(x)$, which maps from a state trajectory $x : \{t\}_{t=0}^{T} \rightarrow X$ to a positive real number. In many cases, $c(x)$ is simply the total time step $T$ to reach the goal. Considering the difficulty in finding the absolute minimal-cost state trajectory, many mobile robots use sampling-based motion planners to find near-optimal solutions [22, 23].

Conversely, in our case of wheeled mobility on vertically challenging terrain, vehicle state $X \subset \mathbb{SE}(3)$ (i.e., translations and rotations along the $\mathbf{x}$, $\mathbf{y}$, and $\mathbf{z}$ axis) with the same input $\mathbf{u}_t = (v_t, \omega_t) \in$

$U \subset \mathbb{R}^2$. The system dynamics enforces that $\mathbf{x}_t$ is always "on top of" a subset of $X_{\text{obs}}$ (i.e., vertically challenging terrain underneath and supporting the robot) or some boundary of $X$ (i.e., on a flat ground) due to gravity, requiring a vehicle-terrain dynamics model (Figure 2 right).

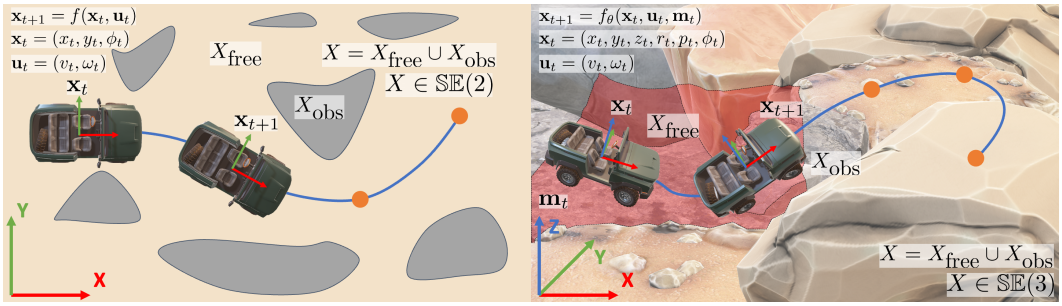


Figure 2: 2D navigation in $\mathbb{SE}(2)$ vs. navigation on vertically challenging terrain in $\mathbb{SE}(3)$.

### 2.2 Vehicle-Terrain Dynamics Model Learning

Compared to the simple 2D vehicle dynamics in $\mathbb{SE}(2)$, our non-rigid vehicle-terrain dynamics on vertically challenging terrain in $\mathbb{SE}(3)$ becomes more difficult to model, considering the complex interaction between the terrain and chassis via the long suspension travel and deflated tire pressure of off-road vehicles to assure adaptivity and traction (Figure 1). Therefore, this work adopts a data-driven approach to learn the vehicle-terrain dynamics model, which can be used to rollout trajectories for subsequent planning.

To be specific, $\mathbf{x}_t = (x_t, y_t, z_t, r_t, p_t, \phi_t)$, where the first and last three denote the translational ($x$, $y$, $z$) and rotational (roll, pitch, yaw) component respectively along the $\mathbf{x}$, $\mathbf{y}$, and $\mathbf{z}$ axis. Note that unlike most 2D navigation problems in which the next vehicle state $\mathbf{x}_{t+1}$ only relies on the current vehicle state $\mathbf{x}_t$ and input $\mathbf{u}_t$ alone, our next vehicle state is additionally affected by the vertically challenging terrain underneath and in front of the vehicle in the current time step, denoted as $\mathbf{m}_t$. Therefore, the forward dynamics on vertically challenging terrain can be formulated as

$$\mathbf{x}_{t+1} = f_\theta(\mathbf{x}_t, \mathbf{u}_t, \mathbf{m}_t), \tag{1}$$

which is parameterized by $\theta$ and will be learned in a data-driven manner. Training data of size $N$ can be collected by driving a wheeled robot on different vertically challenging terrain and recording the current and next state, current terrain, and current input: $\mathcal{D} = \{\langle \mathbf{x}_t, \mathbf{x}_{t+1}, \mathbf{m}_t, \mathbf{u}_t \rangle_{t=1}^{N}\}$. Then we learn $\theta$ by minimizing a supervised loss function:

$$\theta^* = \arg\min_\theta \sum_{(\mathbf{x}_t, \mathbf{x}_{t+1}, \mathbf{m}_t, \mathbf{u}_t) \in \mathcal{D}} \| f_\theta(\mathbf{x}_t, \mathbf{u}_t, \mathbf{m}_t) - \mathbf{x}_{t+1} \|_H, \tag{2}$$

where $||\mathbf{v}||_H = \mathbf{v}^T H \mathbf{v}$ is the norm induced by a positive definite matrix $H$, used to weigh the learning loss of the different dimensions of the vehicle state $\mathbf{x}_t$. The learned vehicle-terrain forward dynamics model, $f_\theta(\cdot, \cdot, \cdot)$, can then be used to rollout future trajectories for minimal-cost planning.

### 2.3 Sampling-Based Receding-Horizon Planning

We adopt a sampling-based receding-horizon planning paradigm, in which the planner first uniformly samples input sequences up until a short horizon $H$, uses the learned model $f_\theta$ to rollout state trajectories, evaluates their cost based on a pre-defined cost function, finds the minimal-cost trajectory, executes the first input, replans, and thus gradually moves the horizon closer to the final goal. In this way, the modeling error can be corrected by frequent replanning. However, an under-actuated wheeled robot, i.e., using $\mathbf{u}_t = (v_t, \omega_t) \in U \subset \mathbb{R}^2$ to actuate $\mathbf{x}_t = (x_t, y_t, z_t, r_t, p_t, \phi_t) \in X \subset \mathbb{SE}(3)$ subject to $f_\theta$, may easily end up in many terminal states outside of $X_{\text{goal}}$, which the vehicle cannot escape and recover from, i.e., rolling over or immobilization (getting stuck) due to excessive roll and pitch angles, irregular terrain geometry, and large height change, e.g.,

on a large rock. Therefore, while our goal is still to minimize the traversal time $T$ leading to $X_{\text{goal}}$, for our receding-horizon planner, we seek to optimize five cost terms on a state trajectory $\mathbf{x}_{0:H} = \{\mathbf{x}_t\}_{t=0}^{H}$, s.t., $\mathbf{x}_{t+1} = f_\theta(\mathbf{x}_t, \mathbf{u}_t, \mathbf{m}_t), \forall t < H$, which starts at the current time 0 and ends at the horizon $H$, to avoid these two types of terminal states on vertical challenging terrain and also move the robot towards the goal:

$$\begin{aligned} c(\mathbf{x}_{0:H}) = w_1 c_{\text{rp}}(\mathbf{x}_{0:H}) + w_2 c_{\text{tg}}(\mathbf{x}_{0:H}) + w_3 c_{\text{hc}}(\mathbf{x}_{0:H}) \\ + w_4 c_{\text{mb}}(\mathbf{x}_{0:H}) + w_5 c_{\text{est}}(x_H), \end{aligned} \tag{3}$$

where $c_{\text{rp}}(\cdot)$, $c_{\text{tg}}(\cdot)$, and $c_{\text{hc}}(\cdot)$ denote the cost corresponding to the robot's (extensive) roll and pitch angle, (irregular) underneath terrain geometry, and (large) terrain height change respectively; $c_{\text{mb}}(\cdot)$ is the cost of moving out of the observable map boundary; $c_{\text{est}}(\cdot)$ is the estimated cost to reach the final goal region $X_{\text{goal}}$ from the state on the horizon $x_H$, which can be computed by the Euclidean distance $c_{\text{est}}(x_H) = ||x_H - x_G||_2$, where $x_G$ is any state inside $X_{\text{goal}}$. $w_1$ to $w_5$ are corresponding weights for the cost terms.

## 3 Experiments and Results

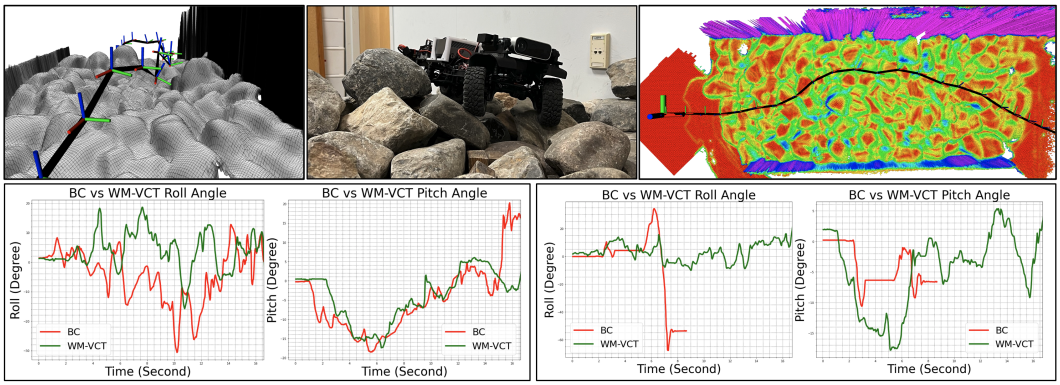


Figure 3: Physical Experiments

Our proposed WM-VCT navigation planner is compared against a Behavior Cloning (BC) baseline [14]. In Figure 3, we show the V6W navigating the testbed (top middle), front (top left) and top (top right) view of the elevation map with the planned 6-DoF vehicle state trajectory, and pitch and roll values in two example environments (bottom left and right). In the first environment, while both BC (red) and WM-VCT (green) succeed, the former experiences larger roll and pitch values; in the second environment, BC (red) fails due to the excessive roll angle around 7.5s, while WM-VCT is able to successfully navigate through.

Table 1 shows our experiment results in three obstacle courses with three difficulty levels, five trials each. In general, our WM-VCT planner achieves better results on both six-wheeled and four-wheeled platforms, compared to BC, the only baseline that can occasionally navigate through, in terms of navigation success rate and average roll and pitch angles. In general, WM-VCT finishes more trials, is slower but more stable, and achieves lower roll and pitch angles overall.

Table 1: Number of successful trials, mean successful traversal time, and average roll/pitch angles.

| | V6W | | V4W | |
|---|---|---|---|---|
| | BC | WM-VCT | BC | WM-VCT |
| Easy | **5**, **15.8s**, 7.3°/7.9° | **5**, 24s, **5.1°**/**7.5°** | **2**, **18.0s**, 9.2°/17.5° | **2**, 27.5s, **5.8°**/**9.5°** |
| Medium | 3, **17.0s**, 9.4°/**8.3°** | **4**, 24.5s, **6.1°**/8.6° | 1, **16.0s**, 12°/**8.5°** | **2**, 32.5s, **7.9°**/11.4° |
| Difficult | 1, **20.0s**, 8.3°/10.7° | **4**, 22.7s, **6.2°**/**7.4°** | N/A | N/A |

# References


[1] IEEE Spectrum. Kiva systems – three engineers, hundreds of robots, one warehouse. https://spectrum.ieee.org/three-engineers-hundreds-of-robots-one-warehouse, 2008. Accessed: 2023-05-16.

[2] iRobot. iRobot – robot vacuum and mop. https://www.irobot.com/, 2023. Accessed: 2023-05-16.

[3] Amazon. Meet scout. https://www.aboutamazon.com/news/transportation/meet-scout, 2023. Accessed: 2023-05-16.

[4] D. Fox, W. Burgard, and S. Thrun. The dynamic window approach to collision avoidance. *IEEE Robotics & Automation Magazine*, 4(1):23–33, 1997.

[5] S. Quinlan and O. Khatib. Elastic bands: Connecting path planning and control. In *[1993] Proceedings IEEE International Conference on Robotics and Automation*, pages 802–807. IEEE, 1993.

[6] C. Rösmann, F. Hoffmann, and T. Bertram. Kinodynamic trajectory optimization and control for car-like robots. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5681–5686. IEEE, 2017.

[7] F. Cordes, C. Oekermann, A. Babu, D. Kuehn, T. Stark, F. Kirchner, and D. Bremen. An active suspension system for a planetary rover. In *Proceedings of the International Symposium on Artificial Intelligence, Robotics and Automation in Space (i-SAIRAS)*, pages 17–19, 2014.

[8] M. R. Islam, F. H. Chowdhury, S. Rezwan, M. J. Ishaque, J. U. Akanda, A. S. Tuhel, and B. B. Riddhe. Novel design and performance analysis of a mars exploration robot: Mars rover mongol pothik. In *2017 Third International Conference on Research in Computational Intelligence and Communication Networks (ICRCICN)*, pages 132–136. IEEE, 2017.

[9] H. Jiang, G. Xu, W. Zeng, F. Gao, and K. Chong. Lateral stability of a mobile robot utilizing an active adjustable suspension. *Applied Sciences*, 9(20):4410, 2019.

[10] Y. Liu and T. Seo. Anyclimb-ii: Dry-adhesive linkage-type climbing robot for uneven vertical surfaces. *Mechanism and Machine Theory*, 124:197–210, 2018.

[11] R. R. Murphy. *Disaster robotics*. MIT press, 2014.

[12] X. Xiao and R. Murphy. A review on snake robot testbeds in granular and restricted maneuverability spaces. *Robotics and Autonomous Systems*, 110:160–172, 2018.

[13] P. McGarey, F. Pomerleau, and T. D. Barfoot. System design of a tethered robotic explorer (trex) for 3d mapping of steep terrain and harsh environments. In *Field and Service Robotics: Results of the 10th International Conference*, pages 267–281. Springer, 2016.

[14] A. Datar, C. Pan, M. Nazeri, and X. Xiao. Toward wheeled mobility on vertically challenging terrain: Platforms, datasets, and algorithms. *arXiv preprint arXiv:2303.00998*, 2023.

[15] X. Xiao, E. Cappo, W. Zhen, J. Dai, K. Sun, C. Gong, M. J. Travers, and H. Choset. Locomotive reduction for snake robots. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3735–3740. IEEE, 2015.

[16] R. Murphy, J. Dufek, T. Sarmiento, G. Wilde, X. Xiao, J. Braun, L. Mullen, R. Smith, S. Allred, J. Adams, et al. Two case studies and gaps analysis of flood assessment for emergency management with small unmanned aerial systems. In *2016 IEEE international symposium on safety, security, and rescue robotics (SSRR)*, pages 54–61. IEEE, 2016.

[17] X. Xiao, J. Dufek, T. Woodbury, and R. Murphy. Uav assisted usv visual navigation for marine mass casualty incident response. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6105–6110. IEEE, 2017.

[18] X. Xiao, J. Dufek, and R. R. Murphy. Autonomous visual assistance for robot operations using a tethered uav. In *Field and Service Robotics: Results of the 12th International Conference*, pages 15–29. Springer, 2021.

[19] R. N. Jazar. *Vehicle dynamics*, volume 1. Springer, 2008.

[20] G. Yan, M. Fang, and J. Xu. Analysis and experiment of time-delayed optimal control for vehicle suspension system. *Journal of Sound and Vibration*, 446:144–158, 2019.

[21] A. A. Aly and F. A. Salem. Vehicle suspension systems control: a review. *International journal of control, automation and systems*, 2(2):46–54, 2013.

[22] S. Karaman and E. Frazzoli. Incremental sampling-based algorithms for optimal motion planning. *Robotics Science and Systems VI*, 104(2), 2010.

[23] S. Karaman, M. R. Walter, A. Perez, E. Frazzoli, and S. Teller. Anytime motion planning using the rrt. In *2011 IEEE international conference on robotics and automation*, pages 1478–1483. IEEE, 2011.